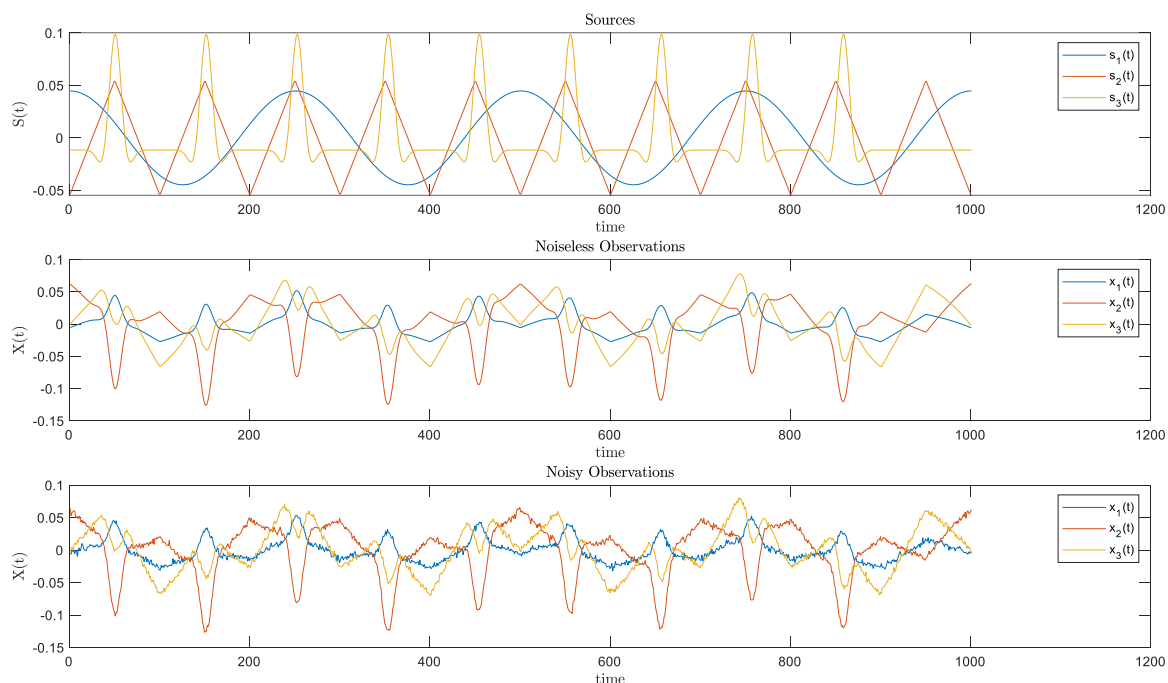


ابتدا ماتریس های منابع و مخلوط کننده و همچنین ماتریس نویز را در متلب وارد می کنیم. سپس با استفاده از $X = AS$ ماتریس مشاهدات را بدست می آوریم. تصویر ۱، منابع و مشاهدات بدون نویز و با نویز را نشان می دهد.



تصویر ۱: منابع و مشاهدات بدون نویز و با نویز

همانطور که در تصویر نیز مشاهده می شود، منابع کاملاً از هم مستقل خطی هستند اما مشاهدات هرکدام از چند منبع ساخته شده اند و به عبارت دیگر ترکیب خطی چند منبع هستند و به همین خاطر دیگر از هم مستقل آماری نیستند.

سوال ۱.

کد مربوط به کمینه سازی D_{KL} با استفاده از روش MSE به صورت زیر است.

```
L = size(X,2);
t = 1:L;
K = @(Y) [ones(1,L);Y;Y.^2;Y.^3;Y.^4;Y.^5];
dK = @(Y) [zeros(1,L);ones(1,L);2*Y;3*Y.^2;4*Y.^3;5*Y.^4];
% init
B=[0.321,0.532,0.533;
    0.227,0.41,0.282;
    0.321,0.821,0.81];
B = B./sqrt(sum(B.^2,2));
mu = 0.01;
Y = B * X;
ITR = 1000;
```

```

Score = zeros(1, ITR);
for itr = 1:ITR
    K1 = K(Y(1,:));
    K2 = K(Y(2,:));
    K3 = K(Y(3,:));
    Theta1 = (K1*K1.'/L)\mean(dK(Y(1,:)),2);
    Theta2 = (K2*K2.'/L)\mean(dK(Y(2,:)),2);
    Theta3 = (K3*K3.'/L)\mean(dK(Y(3,:)),2);
    Psi1 = Theta1.' * K1;
    Psi2 = Theta2.' * K2;
    Psi3 = Theta3.' * K3;
    grad_f = ([Psi1*X.' ; Psi2*X.' ; Psi3*X.']/L) - (inv(B)).';
    B = B - mu * grad_f;
    B = B./sqrt(sum(B.^2,2));
    S_hat = B * X;
    Shat_ = S_hat;
    S_ = S;
    [~,r1] = max(abs(Shat_(1,:)*S_'));
    S_(r1,:) = 0;
    [~,r2] = max(abs(Shat_(2,:)*S_'));
    S_(r2,:) = 0;
    [~,r3] = max(abs(Shat_(3,:)*S_'));
    S_(r3,:) = 0;
    S_hat(r1,:) = Shat_(1,:);
    S_hat(r2,:) = Shat_(2,:);
    S_hat(r3,:) = Shat_(3,:);
    S_hat(1,:) = S_hat(1,:) * max(abs(S(1,:))) / max(abs(S_hat(1,:)));
    S_hat(2,:) = - S_hat(2,:) * max(abs(S(2,:))) / max(abs(S_hat(2,:)));
    S_hat(3,:) = - S_hat(3,:) * max(abs(S(3,:))) / max(abs(S_hat(3,:)));
    Score(itr) = norm(grad_f)^2;
    Y = B * X;
end
figure
plot(Score);
xlim([0 100]);
title('Convergence Diagram (Kurnel)','Interpreter','latex');
xlabel('iteration','Interpreter','latex');
ylabel('Error','Interpreter','latex');
E = norm(S - S_hat,'fro')^2/norm(S,'fro');
fprintf('Kurnel : E = %f\n',E(end));
figure
subplot(3,1,1);
plot(t,S(1,:), 'b',t,S_hat(1,:), 'r');
legend('s_1(t)', 'shat_1(t)');
title('Source 1 (Kurnel)','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_1(t)$','Interpreter','latex');
subplot(3,1,2);
plot(t,S(2,:), 'b',t,S_hat(2,:), 'r');
legend('s_2(t)', 'shat_2(t)');
title('Source 2 (Kurnel)','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_2(t)$','Interpreter','latex');
subplot(3,1,3);
plot(t,S(3,:), 'b',t,S_hat(3,:), 'r');
legend('s_3(t)', 'shat_3(t)');
title('Source 3 (Kurnel)','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_3(t)$','Interpreter','latex');
Permutation_kurnel = B * A

```

نتایج :

۱. **ماتریس $Permutation$** : این ماتریس را به صورت $B \times A$ بدست می آوریم. تصویر ۲ این ماتریس را نشان می دهد.

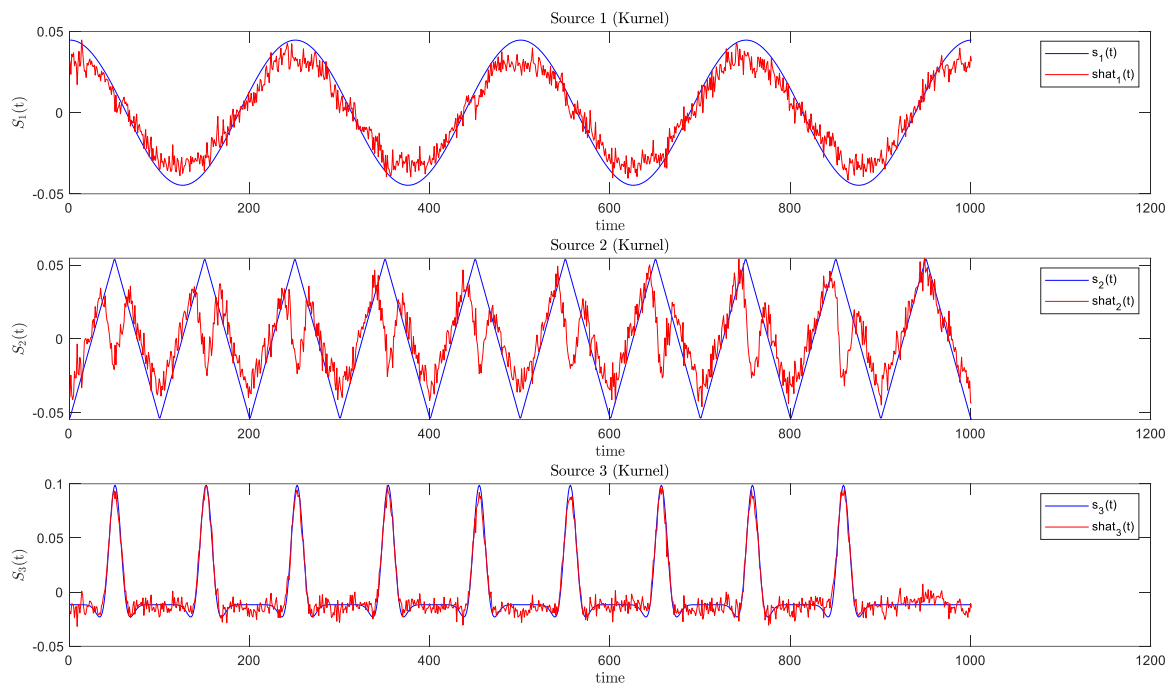
بخاطر وجود نویز مقادیر کمی از حالت ایده آل فاصله دارد اما می توان عنصر غیر صفر را تشخیص داد.

Permutation_kernel =

```
0.4374    0.0420   -0.0426
0.0127   -0.3192    0.2220
0.0134   -0.0667   -0.4679
```

تصویر ۲: ماتریس *Permutation*

۲. مقدار خطا: برای رفع ابهام ترتیب و scale از منابع اصلی استفاده می‌کنیم. تصویر ۳، منابع تخمین زده شده با این روش را نشان می‌دهد.



تصویر ۳: منابع تخمین زده شده

در نهایت مقدار خطای E به صورت زیر تعیین می‌شود.

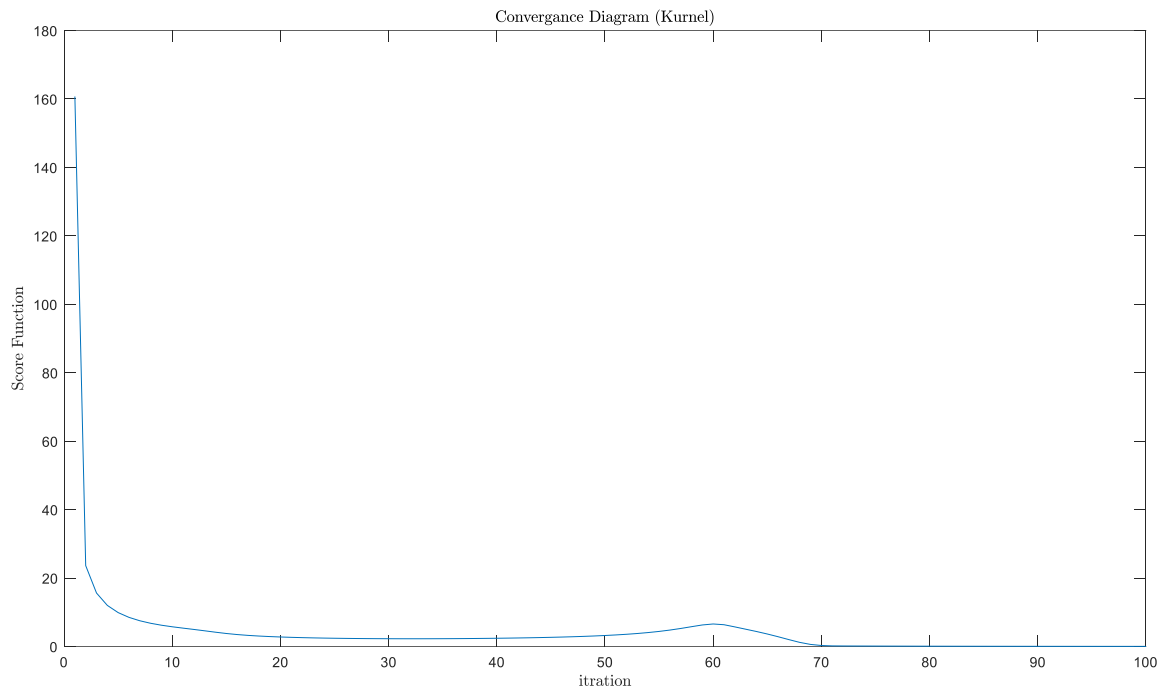
$$E = \frac{\|S - \hat{S}\|}{\|S\|}$$

تصویر ۴ مقدار خطا برای این روش را نشان می‌دهد.

Kurnel : $E = 0.369812$

تصویر ۴: مقدار E

۳. نمودار همگرایی تابع هدف: تصویر ۵، نمودار همگرایی تابع هدف برای این روش را نشان می‌دهد.



تصویر ۵: نمودار همگرایی تابع هدف

سوال ۲.

کد مربوط به کمینه سازی D_{KL} با استفاده از روش *Deflation* به صورت زیر است.

```
% init
B = [0.321, 0.532, 0.533;
     0.227, 0.41, 0.282;
     0.321, 0.821, 0.81];
B = B./sqrt(sum(B.^2,2));
% Whitening
Rx = X * X.';
[U,D] = eig(Rx);
W = (D^(-1/2))*U';
Z = W * X;
Y = B * Z;
mu = 0.01;
for itr = 1:1000
    K1 = K(Y(1,:));
    K2 = K(Y(2,:));
    K3 = K(Y(3,:));
    Theta1 = (K1*K1.'/L)\mean(dK(Y(1,:)),2);
    Theta2 = (K2*K2.'/L)\mean(dK(Y(2,:)),2);
    Theta3 = (K3*K3.'/L)\mean(dK(Y(3,:)),2);
    Psi1 = Theta1.' * K1;
    Psi2 = Theta2.' * K2;
    Psi3 = Theta3.' * K3;
    grad_H = [Psi1*Z.' ; Psi2*Z.' ; Psi3*Z.'];
    b1 = B(1,:).';
    b2 = B(2,:).';
    b3 = B(3,:).';
    b1 = b1 - mu * (grad_H(1,:)).';
    b1 = b1 / norm(b1);
    b2 = b2 - mu * (grad_H(2,:)).';
    b2 = (eye(3) - b1*b1.') * b2;
    b2 = b2 / norm(b2);
    b3 = b3 - mu * (grad_H(3,:)).';
```

```

b3 = (eye(3) - [b1,b2]*[b1,b2].') * b3;
b3 = b3 / norm(b3);
B = [b1.';b2.';b3.'];
S_hat = B * X;
Shat_ = S_hat;
S_ = S;
[~,r1] = max(abs(Shat_(1,:)*S_'));
S_(r1,:) = 0;
[~,r2] = max(abs(Shat_(2,:)*S_'));
S_(r2,:) = 0;
[~,r3] = max(abs(Shat_(3,:)*S_'));
S_(r3,:) = 0;
S_hat(r1,:) = Shat_(1,:);
S_hat(r2,:) = Shat_(2,:);
S_hat(r3,:) = Shat_(3,:);
S_hat(1,:) = - S_hat(1,:) * max(abs(S(1,:))) / max(abs(S_hat(1,:)));
S_hat(2,:) = S_hat(2,:) * max(abs(S(2,:))) / max(abs(S_hat(2,:)));
S_hat(3,:) = - S_hat(3,:) * max(abs(S(3,:))) / max(abs(S_hat(3,:)));
Score(itr) = norm(grad_H) / L;
Y = B * Z;
end
figure
plot(Score);
xlim([0 100]);
title('Convergence Diagram (Deflation)','Interpreter','latex');
xlabel('Iteration','Interpreter','latex');
ylabel('Score Function','Interpreter','latex');
E = norm(S - S_hat,'fro')^2/norm(S,'fro');
fprintf("Deflation : E = %f\n",E(end));
figure
subplot(3,1,1);
plot(t,S(1,:), 'b',t,S_hat(1,:), 'r');
legend('s_1(t)', 'shat_1(t)');
title('Source 1 (Deflation)','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_1(t)$','Interpreter','latex');
subplot(3,1,2);
plot(t,S(2,:), 'b',t,S_hat(2,:), 'r');
legend('s_2(t)', 'shat_2(t)');
title('Source 2 (Deflation)','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_2(t)$','Interpreter','latex');
subplot(3,1,3);
plot(t,S(3,:), 'b',t,S_hat(3,:), 'r');
legend('s_3(t)', 'shat_3(t)');
title('Source 3 (Deflation)','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_3(t)$','Interpreter','latex');
Permutation_deflation = B * W * A

```

نتایج :

۱. **ماتریس *Permutation*** : این ماتریس را به صورت $B \times W \times A$ بدست می‌آوریم. تصویر ۶ این ماتریس را نشان

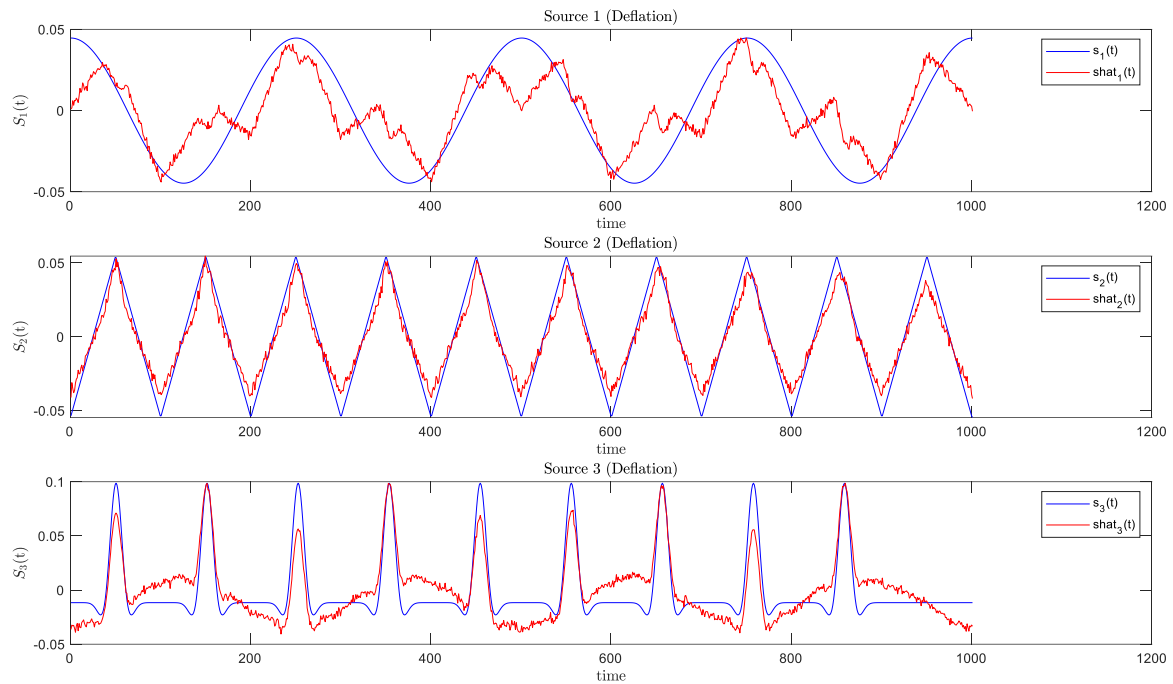
می‌دهد.

Permutation_deflation =

0.4437	0.8204	-0.9662
-0.7850	0.6670	-0.2193
-0.4105	-0.4656	-0.5668

تصویر ۶: ماتریس *Permutation*

۲. مقدار خطا: تصویر ۷، منابع تخمین زده شده با این روش را نشان می‌دهد.



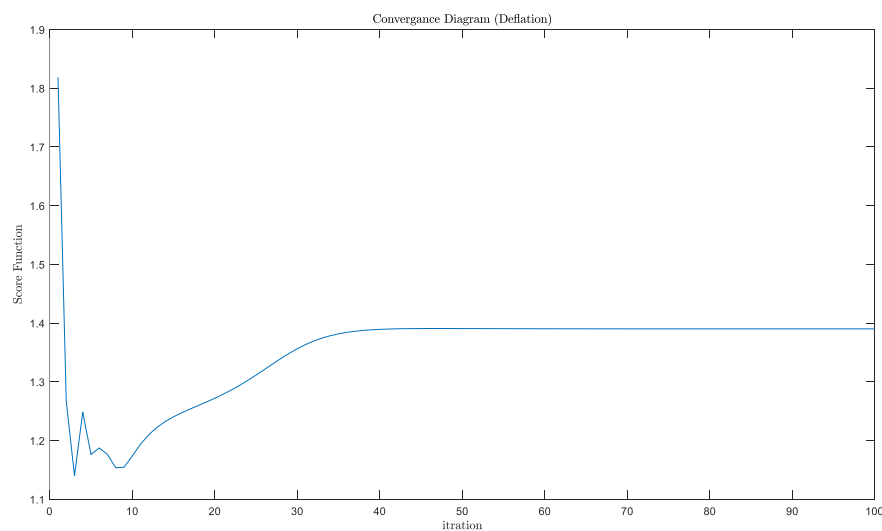
تصویر ۷: منابع تخمین زده شده

تصویر ۴ مقدار خطا برای این روش را نشان می‌دهد.

Deflation : $E = 0.432939$

تصویر ۸: مقدار E

۳. نمودار همگرایی تابع هدف: تصویر ۵، نمودار همگرایی تابع هدف برای این روش را نشان می‌دهد.



تصویر ۹: نمودار همگرایی تابع هدف

سوال ۳.

کد مربوط به کمینه سازی D_{KL} با استفاده از روش *Equivariant* به صورت زیر است.

```
% init
B = [0.321, 0.532, 0.533;
      0.227, 0.41, 0.282;
      0.321, 0.821, 0.81];
mu = 0.05;
Y = B * X;
ITR = 1000;
E = zeros(1, ITR);
for itr = 1:ITR
    K1 = K(Y(1,:));
    K2 = K(Y(2,:));
    K3 = K(Y(3,:));
    Theta1 = (K1*K1.' / L) \ mean(dK(Y(1,:)), 2);
    Theta2 = (K2*K2.' / L) \ mean(dK(Y(2,:)), 2);
    Theta3 = (K3*K3.' / L) \ mean(dK(Y(3,:)), 2);
    Psi1 = Theta1.' * K1;
    Psi2 = Theta2.' * K2;
    Psi3 = Theta3.' * K3;
    grad_f = ([Psi1*X.' ; Psi2*X.' ; Psi3*X.'] / L) - (inv(B)).';
    B = (eye(3) - mu * grad_f * B.') * B;
    B = B ./ sqrt(sum(B.^2, 2));
    S_hat = B * X;
    Shat_ = S_hat;
    S_ = S;
    [~, r1] = max(abs(Shat_(1,:) * S_'));
    S_(r1,:) = 0;
    [~, r2] = max(abs(Shat_(2,:) * S_'));
    S_(r2,:) = 0;
    [~, r3] = max(abs(Shat_(3,:) * S_'));
    S_(r3,:) = 0;
    S_hat(1,:) = S_hat(1,:) * max(abs(S(1,:))) / max(abs(S_hat(1,:)));
    S_hat(2,:) = -S_hat(2,:) * max(abs(S(2,:))) / max(abs(S_hat(2,:)));
    S_hat(3,:) = -S_hat(3,:) * max(abs(S(3,:))) / max(abs(S_hat(3,:)));
    Score(itr) = norm(grad_f)^2;
    Y = B * X;
end
figure
plot(Score);
xlim([0 100]);
title('Convergence Diagram (Equivariant)', 'Interpreter', 'latex');
xlabel('iteration', 'Interpreter', 'latex');
ylabel('Score Function', 'Interpreter', 'latex');
E = norm(S - S_hat, 'fro')^2 / norm(S, 'fro');
fprintf('Equivariant : E = %f\n', E(end));
figure
subplot(3,1,1);
plot(t, S(1,:), 'b', t, S_hat(1,:), 'r');
legend('s_1(t)', 'shat_1(t)');
title('Source 1 (Equivariant)', 'Interpreter', 'latex');
xlabel('time', 'Interpreter', 'latex');
ylabel('$S_1(t)$', 'Interpreter', 'latex');
subplot(3,1,2);
plot(t, S(2,:), 'b', t, S_hat(2,:), 'r');
legend('s_2(t)', 'shat_2(t)');
title('Source 2 (Equivariant)', 'Interpreter', 'latex');
xlabel('time', 'Interpreter', 'latex');
ylabel('$S_2(t)$', 'Interpreter', 'latex');
subplot(3,1,3);
plot(t, S(3,:), 'b', t, S_hat(3,:), 'r');
legend('s_3(t)', 'shat_3(t)');
title('Source 3 (Equivariant)', 'Interpreter', 'latex');
xlabel('time', 'Interpreter', 'latex');
ylabel('$S_3(t)$', 'Interpreter', 'latex');
Permutation_equivariant = B * A
```

نتایج :

۱. **ماتریس $Permutation$** : این ماتریس را به صورت $B \times A$ بدست می‌آوریم. تصویر ۱۰ این ماتریس را نشان می‌دهد.

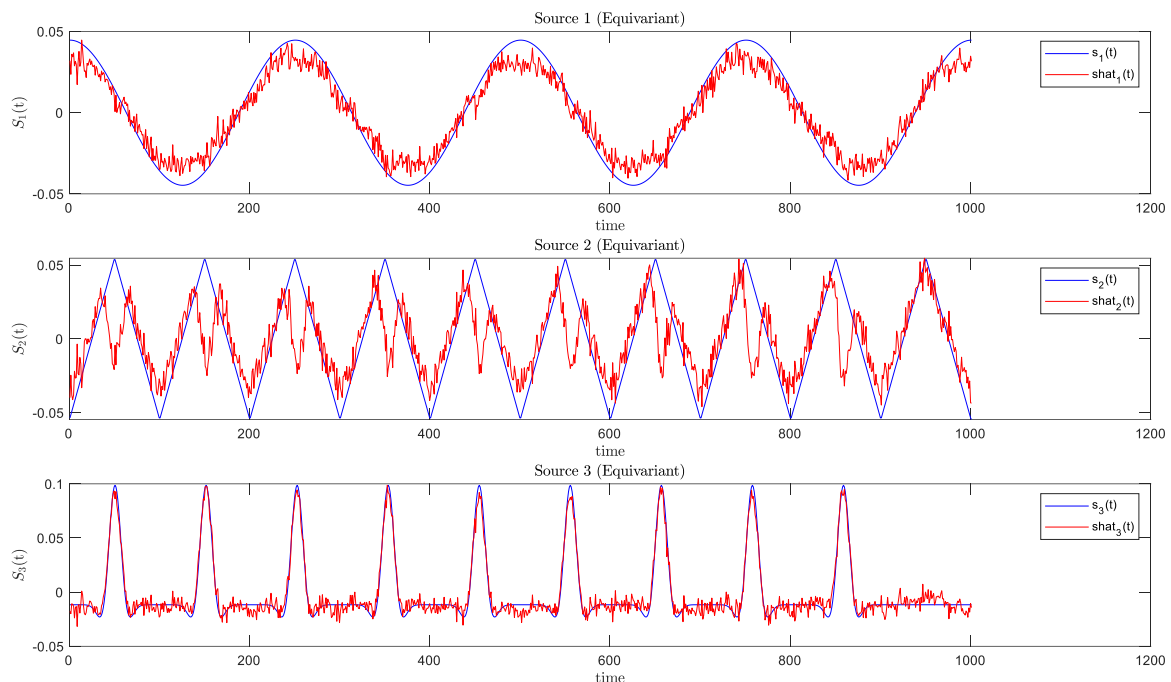
بخاطر وجود نویز مقادیر کمی از حالت ایده آل فاصله دارد اما می‌توان عنصر غیر صفر را تشخیص داد.

Permutation_equivariant =

0.4374	0.0420	-0.0426
0.0127	-0.3192	0.2220
0.0134	-0.0667	-0.4679

تصویر ۱۰: ماتریس $Permutation$

۲. **مقدار خطا**: تصویر ۱۱، منابع تخمین زده شده با این روش را نشان می‌دهد.



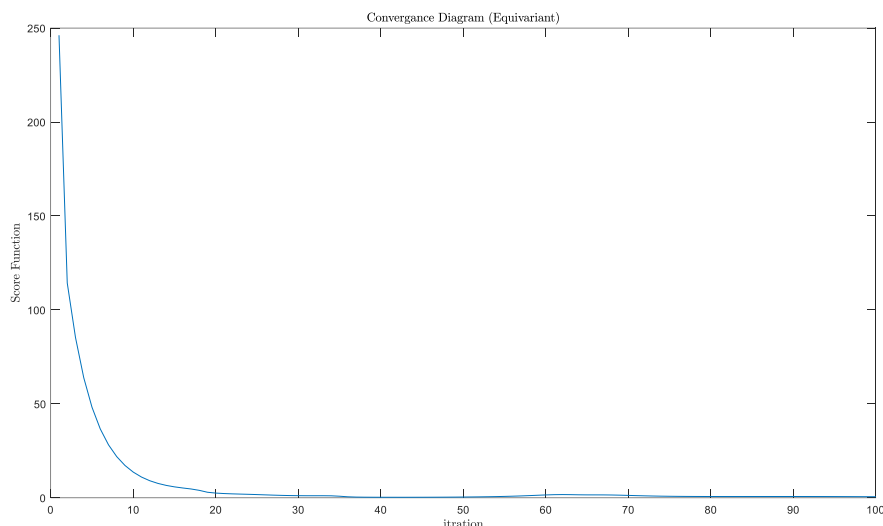
تصویر ۱۱: منابع تخمین زده شده

تصویر ۴ مقدار خطا برای این روش را نشان می‌دهد.

Equivariant : $E = 0.369812$

تصویر ۱۲: مقدار E

۳. **نمودار همگرایی تابع هدف**: تصویر ۱۳، نمودار همگرایی تابع هدف برای این روش را نشان می‌دهد.



تصویر ۱۳: نمودار همگرایی تابع هدف

سوال ۴.

- از نظر سرعت همگرایی :

Deflation < **Kurnel** = **Equivariant**

- از نظر کیفیت جداسازی:

Deflation < **Kurnel** = **Equivariant**

سوال ۵.

کد مربوط به بیشینه سازی *Kurt* با استفاده از روش *Deflation* به صورت زیر است.

```
% init
B = [0.321, 0.532, 0.533;
     0.227, 0.41, 0.282;
     0.321, 0.821, 0.81];
B = B ./ sqrt(sum(B.^2, 2));
% Whitening
Rx = X * X.';
[U, D] = eig(Rx);
W = (D^(-1/2)) * U';
Z = W * X;
Y = B * Z;
mu = 0.01;
for itr = 1:1000
    y1 = Y(1, :);
    b1 = B(1, :).';
    kurt1 = mean(y1.^4) - 3*(mean(y1.^2))^2;
    grad_f1 = sign(kurt1) * (mean((ones(3, 1) * y1.^3) .* Z, 2) - 3*b1);
    b1 = b1 + mu * grad_f1;
    b1 = b1 / norm(b1);

    y2 = Y(2, :);
    b2 = B(2, :).';
```

```

kurty2 = mean(y2.^4) - 3*(mean(y2.^2))^2;
grad_f2 = sign(kurty2)*(mean((ones(3,1)*y2.^3).*Z,2)-3*b2);
b2 = -b2 + mu * grad_f2;
b2 = (eye(3) - b1*b1.') * b2;
b2 = b2 / norm(b2);

y3 = Y(3,:);
b3 = B(3,:).';
kurty3 = mean(y3.^4) - 3*(mean(y3.^2))^2;
grad_f3 = sign(kurty3)*(mean((ones(3,1)*y3.^3).*Z,2)-3*b3);
b3 = -b3 + mu * grad_f3;
b3 = (eye(3) - [b1,b2]*[b1,b2].') * b3;
b3 = b3 / norm(b3);

B = [b1.';b2.';b3.'];
S_hat = B * X;
Shat_ = S_hat;
S_ = S;
[~,r1] = max(abs(Shat_(1,:)*S_'));
S_(r1,:) = 0;
[~,r2] = max(abs(Shat_(2,:)*S_'));
S_(r2,:) = 0;
[~,r3] = max(abs(Shat_(3,:)*S_'));
S_(r3,:) = 0;
S_hat(r1,:) = Shat_(1,:);
S_hat(r2,:) = Shat_(2,:);
S_hat(r3,:) = Shat_(3,:);
S_hat(1,:) = S_hat(1,:) * max(abs(S(1,:))) / max(abs(S_hat(1,:)));
S_hat(2,:) = -S_hat(2,:) * max(abs(S(2,:))) / max(abs(S_hat(2,:)));
S_hat(3,:) = -S_hat(3,:) * max(abs(S(3,:))) / max(abs(S_hat(3,:)));
KURTY = mean(Y.^4,2) - 3*(mean(Y.^2,2)).^2;
Score(itr) = norm(KURTY) / L;
Y = B * Z;
end
figure
plot(Score);
xlim([0 100]);
title('Convergnace Diagram (Kurt)','Interpreter','latex');
xlabel('itration','Interpreter','latex');
ylabel('Score Function','Interpreter','latex');
E = norm(S - S_hat,'fro')^2/norm(S,'fro');
fprintf('Kurt (deflation) : E = %f\n",E(end));
figure
subplot(3,1,1);
plot(t,S(1,:), 'b',t,S_hat(1,:), 'r');
legend('s_1(t)', 'shat_1(t)');
title('Source 1 (Kurt)','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_1(t)$','Interpreter','latex');
subplot(3,1,2);
plot(t,S(2,:), 'b',t,S_hat(2,:), 'r');
legend('s_2(t)', 'shat_2(t)');
title('Source 2 (Kurt)','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_2(t)$','Interpreter','latex');
subplot(3,1,3);
plot(t,S(3,:), 'b',t,S_hat(3,:), 'r');
legend('s_3(t) (Kurt)', 'shat_3(t)');
title('Source 3','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_3(t)$','Interpreter','latex');
Permutation_kurt = B * W * A

```

نتایج :

۱. **ماتریس *Permutation*** : این ماتریس را به صورت $B \times W \times A$ بدست می‌آوریم. تصویر ۱۴ این ماتریس را نشان

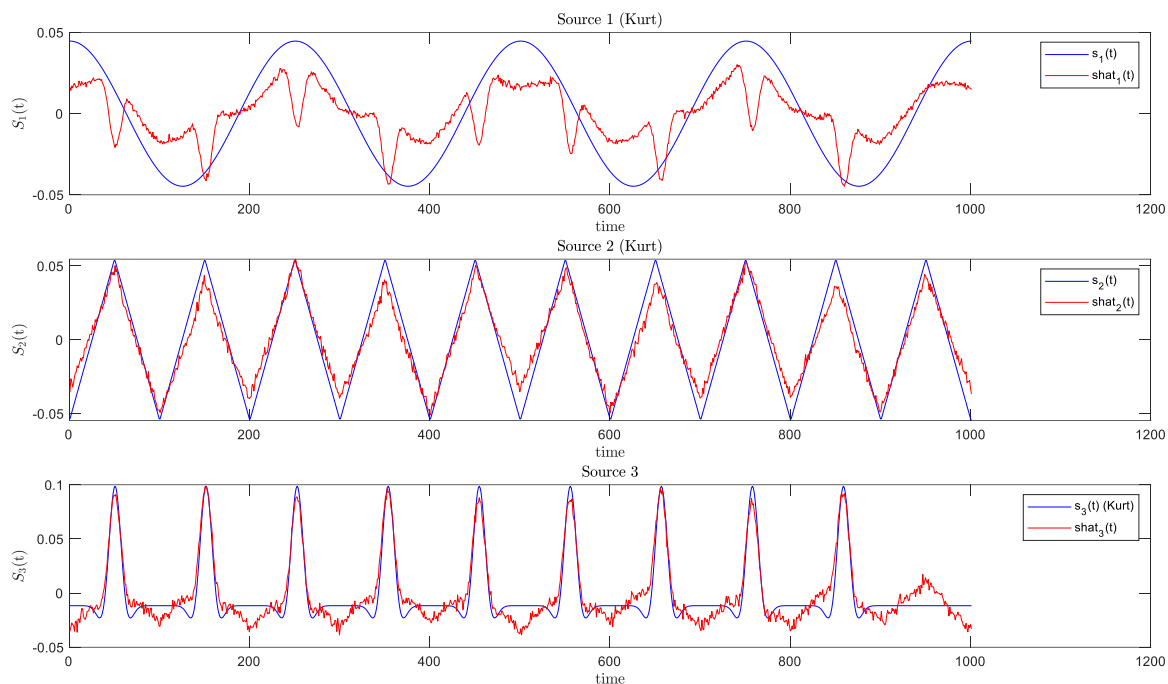
می‌دهد. بخاطر وجود نویز مقادیر کمی از حالت ایده آل فاصله دارد اما می‌توان عنصر غیر صفر را تشخیص داد.

Permutation_kurt =

0.9640	0.0650	-0.2650
0.1785	0.6090	0.5080
-0.1427	0.9797	-0.9873

تصویر ۱۴ : ماتریس *Permutation*

۲. مقدار خطا : تصویر ۱۵ ، منابع تخمین زده شده با این روش را نشان می‌دهد.



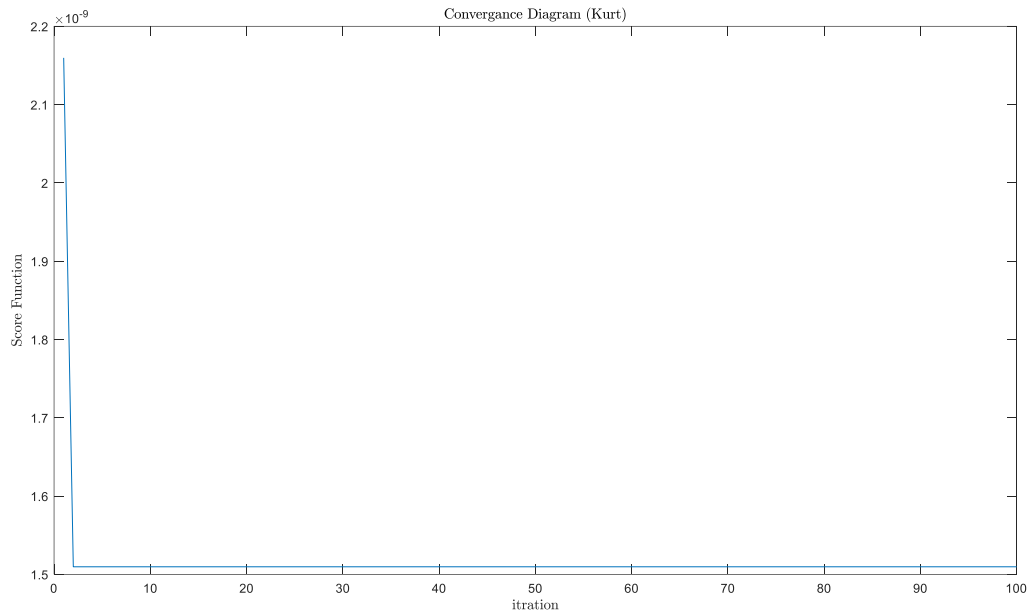
تصویر ۱۵ : منابع تخمین زده شده

تصویر ۱۶ مقدار خطا برای این روش را نشان می‌دهد.

Kurt (deflation) : $E = 0.362806$

تصویر ۱۶ : مقدار E

۳. نمودار همگرایی تابع هدف : تصویر ۱۷ ، نمودار همگرایی تابع هدف برای این روش را نشان می‌دهد.



تصویر ۱۷: نمودار همگرایی تابع هدف

سوال ۶.

کد مربوط به کمینه سازی $Kurt$ با استفاده از روش $fixed - point$ به صورت زیر است.

```
% init
B = [0.321, 0.532, 0.533;
      0.227, 0.41, 0.282;
      0.321, 0.821, 0.81];
B = B./sqrt(sum(B.^2,2));
% Whitening
Rx = X * X.';
[U,D] = eig(Rx);
W = (D^(-1/2))*U';
Z = W * X;
Y = B * Z;
mu = 0.01;
for itr = 1:1000
    y1 = Y(1,:);
    b1 = B(1,:).';
    b1 = mean((ones(3,1)*(y1.^3).*Z),2) - 3*b1;
    b1 = b1 / norm(b1);

    y2 = Y(2,:);
    b2 = B(2,:).';
    b2 = mean((ones(3,1)*(y2.^3).*Z),2) - 3*b2;
    b2 = (eye(3) - b1*b1.') * b2;
    b2 = b2 / norm(b2);

    y3 = Y(3,:);
    b3 = B(3,:).';
    b3 = mean((ones(3,1)*(y3.^3).*Z),2) - 3*b3;
    b3 = (eye(3) - [b1,b2]*[b1,b2].') * b3;
    b3 = b3 / norm(b3);

    B = [b1.';b2.';b3.'];
    S_hat = B * X;
    Shat_ = S_hat;
    S_ = S;
    [~,r1] = max(abs(Shat_(1,:)*S_'));
```

```

S_(r1,:) = 0;
[~,r2] = max(abs(Shat_(2,:)*S_'));
S_(r2,:) = 0;
[~,r3] = max(abs(Shat_(3,:)*S_'));
S_(r3,:) = 0;
S_hat(r1,:) = Shat_(1,:);
S_hat(r2,:) = Shat_(2,:);
S_hat(r3,:) = Shat_(3,:);
S_hat(1,:) = S_hat(1,:) * max(abs(S(1,:))) / max(abs(S_hat(1,:)));
S_hat(2,:) = -S_hat(2,:) * max(abs(S(2,:))) / max(abs(S_hat(2,:)));
S_hat(3,:) = -S_hat(3,:) * max(abs(S(3,:))) / max(abs(S_hat(3,:)));
KURTY = mean(Y.^4,2) - 3*(mean(Y.^2,2)).^2;
Score(itr) = norm(KURTY) / L;
Y = B * Z;
end
figure
plot(Score);
xlim([0 100]);
title('Convergence Diagram (Fixed-Point)','Interpreter','latex');
xlabel('Iteration','Interpreter','latex');
ylabel('Score Function','Interpreter','latex');
fprintf('Fixed-Point : E = %f\n',E(end));
figure
subplot(3,1,1);
plot(t,S(1,:), 'b',t,S_hat(1,:), 'r');
legend('s_1(t)', 'shat_1(t)');
title('Source 1 (Fixed-Point)','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_1(t)$','Interpreter','latex');
subplot(3,1,2);
plot(t,S(2,:), 'b',t,S_hat(2,:), 'r');
legend('s_2(t)', 'shat_2(t)');
title('Source 2 (Fixed-Point)','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_2(t)$','Interpreter','latex');
subplot(3,1,3);
plot(t,S(3,:), 'b',t,S_hat(3,:), 'r');
legend('s_3(t) (Fixed-Point)', 'shat_3(t)');
title('Source 3','Interpreter','latex');
xlabel('time','Interpreter','latex');
ylabel('$S_3(t)$','Interpreter','latex');
Permutation_Fixed_Point = B * W * A

```

نتایج :

۱. **ماتریس *Permutation*** : این ماتریس را به صورت $B \times W \times A$ بدست می‌آوریم. تصویر ۱۸ این ماتریس را نشان

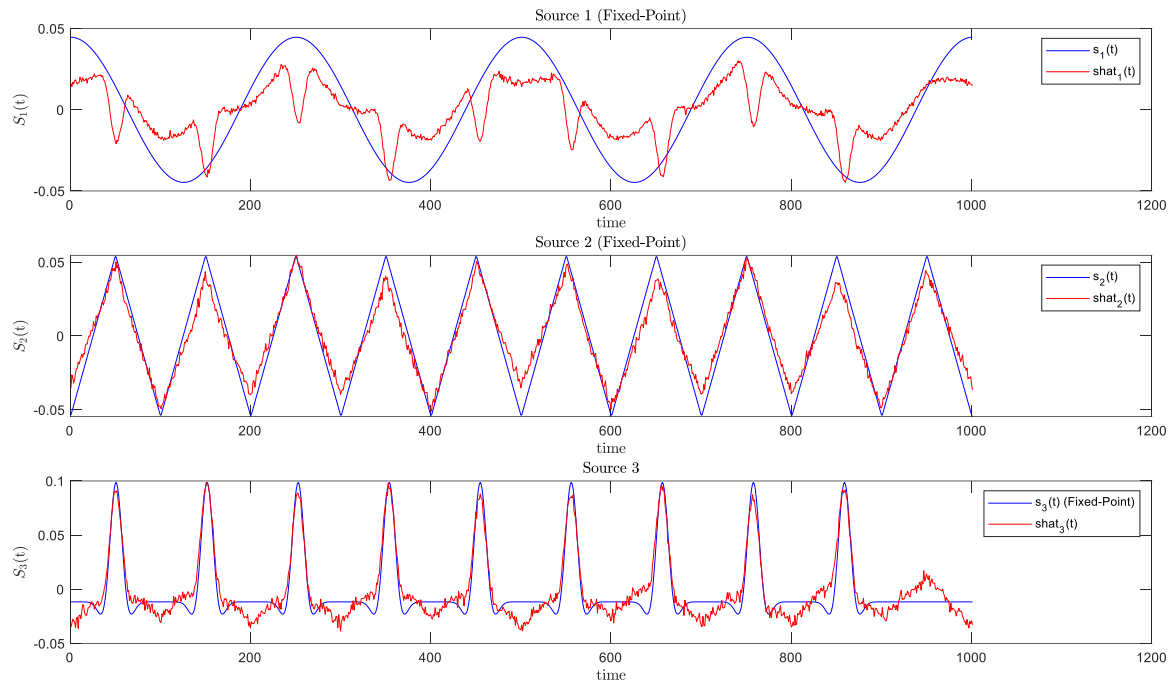
می‌دهد. بخاطر وجود نویز مقادیر کمی از حالت ایده آل فاصله دارد اما می‌توان عنصر غیر صفر را تشخیص داد.

Permutation_Fixed_Point =

0.9641	0.0651	-0.2649
0.1783	0.6094	0.5076
-0.1427	0.9794	-0.9875

تصویر ۱۸ : ماتریس *Permutation*

۲. **مقدار خطا** : تصویر ۱۹ ، منابع تخمین زده شده با این روش را نشان می‌دهد.



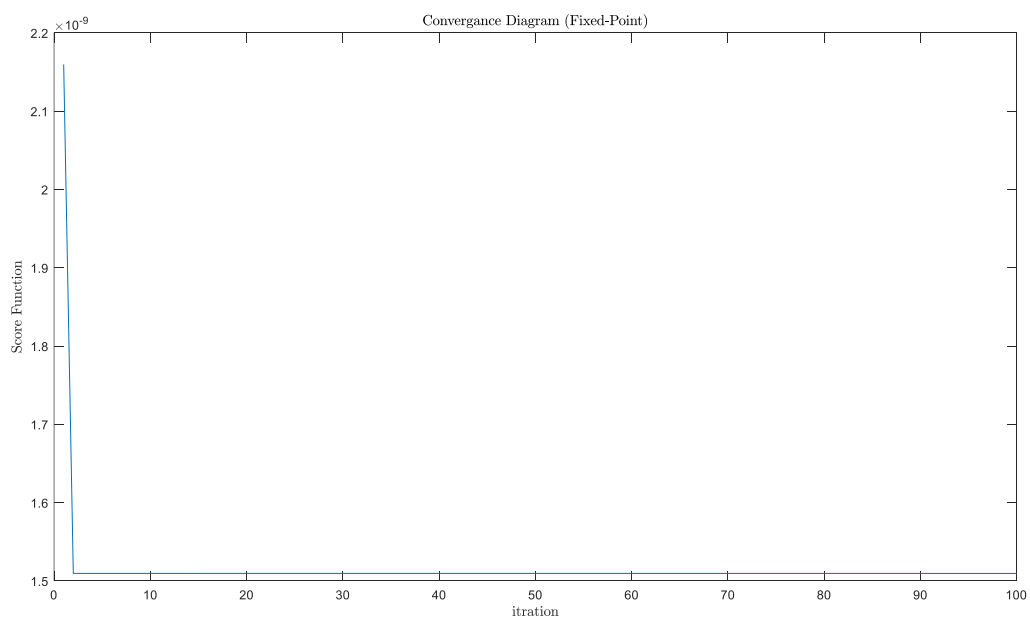
تصویر ۱۹ : منابع تخمین زده شده

تصویر ۲۰ مقدار خطا برای این روش را نشان می‌دهد.

Fixed-Point : $E = 0.362806$

تصویر ۲۰ : مقدار E

۳. نمودار همگرایی تابع هدف : تصویر ۲۱ ، نمودار همگرایی تابع هدف برای این روش را نشان می‌دهد.



تصویر ۲۱ : نمودار همگرایی تابع هدف

سوال ۷.

کد مربوط به روش $G.P$ به صورت زیر است.

```
% init
B = [0.321, 0.532, 0.533;
      0.227, 0.41, 0.282;
      0.321, 0.821, 0.81];
B = B./sqrt(sum(B.^2,2));
v = randn(1,L);
% Whitening
Rx = X * X.';
[U,D] = eig(Rx);
W = (D^(-1/2))*U';
Z = W * X;
Y = B * Z;
mu = 0.01;
for itr = 1:1000
    Gv = -exp(-v.^2/2);
    y1 = Y(1,:);
    b1 = B(1,:).';
    Gy1 = -exp(-y1.^2/2);
    gy1 = -y1 .* Gy1;
    fb1 = (mean(Gy1) - mean(Gv))^2;
    grad_f1 = fb1*(mean(ones(3,1)*gy1).*Z,2);
    b1 = b1 + mu * grad_f1;
    b1 = b1 / norm(b1);

    y2 = Y(2,:);
    b2 = B(2,:).';
    Gy2 = -exp(-y2.^2/2);
    gy2 = -y2 .* Gy2;
    fb2 = (mean(Gy2) - mean(Gv))^2;
    grad_f2 = fb2*(mean(ones(3,1)*gy2).*Z,2);
    b2 = b2 + mu * grad_f2;
    b2 = (eye(3) - b1*b1.') * b2;
    b2 = b2 / norm(b2);

    y3 = Y(3,:);
    b3 = B(3,:).';
    Gy3 = -exp(-y3.^2/2);
    gy3 = -y3 .* Gy3;
    fb3 = (mean(Gy3) - mean(Gv))^2;
    grad_f3 = fb3*(mean(ones(3,1)*gy3).*Z,2);
    b3 = b3 + mu * grad_f3;
    b3 = (eye(3) - [b1,b2]*[b1,b2].') * b3;
    b3 = b3 / norm(b3);

    B = [b1.';b2.';b3.'];
    S_hat = B * X;
    Shat_ = S_hat;
    S_ = S;
    [~,r1] = max(abs(Shat_(1,:)*S_'));
    S_(r1,:) = 0;
    [~,r2] = max(abs(Shat_(2,:)*S_'));
    S_(r2,:) = 0;
    [~,r3] = max(abs(Shat_(3,:)*S_'));
    S_(r3,:) = 0;
    S_hat(r1,:) = Shat_(1,:);
    S_hat(r2,:) = Shat_(2,:);
    S_hat(r3,:) = Shat_(3,:);
    S_hat(1,:) = S_hat(1,:) * max(abs(S(1,:))) / max(abs(S_hat(1,:)));
    S_hat(2,:) = -S_hat(2,:) * max(abs(S(2,:))) / max(abs(S_hat(2,:)));
    S_hat(3,:) = -S_hat(3,:) * max(abs(S(3,:))) / max(abs(S_hat(3,:)));
    KURTY = mean(Y.^4,2) - 3*(mean(Y.^2,2)).^2;
    Score(itr) = norm(KURTY) / L;
    Y = B * Z;
end
figure
```

```

plot(Score);
xlim([0 100]);
title('Convergence Diagram (G.P)', 'Interpreter', 'latex');
xlabel('Iteration', 'Interpreter', 'latex');
ylabel('Score Function', 'Interpreter', 'latex');
E = norm(S - S_hat, 'fro')^2/norm(S, 'fro');
fprintf("G.P : E = %f\n", E(end));
figure
subplot(3,1,1);
plot(t, S(1,:), 'b', t, S_hat(1,:), 'r');
legend('s_1(t)', 'shat_1(t)');
title('Source 1 (G.P)', 'Interpreter', 'latex');
xlabel('time', 'Interpreter', 'latex');
ylabel('$S_1(t)$', 'Interpreter', 'latex');
subplot(3,1,2);
plot(t, S(2,:), 'b', t, S_hat(2,:), 'r');
legend('s_2(t)', 'shat_2(t)');
title('Source 2 (G.P)', 'Interpreter', 'latex');
xlabel('time', 'Interpreter', 'latex');
ylabel('$S_2(t)$', 'Interpreter', 'latex');
subplot(3,1,3);
plot(t, S(3,:), 'b', t, S_hat(3,:), 'r');
legend('s_3(t) (G.P)', 'shat_3(t)');
title('Source 3', 'Interpreter', 'latex');
xlabel('time', 'Interpreter', 'latex');
ylabel('$S_3(t)$', 'Interpreter', 'latex');
Permutation_GP = B * W * A

```

نتایج :

۱. **ماتریس $Permutation$** : این ماتریس را به صورت $B \times W \times A$ بدست می آوریم. تصویر ۲۲ این ماتریس را نشان

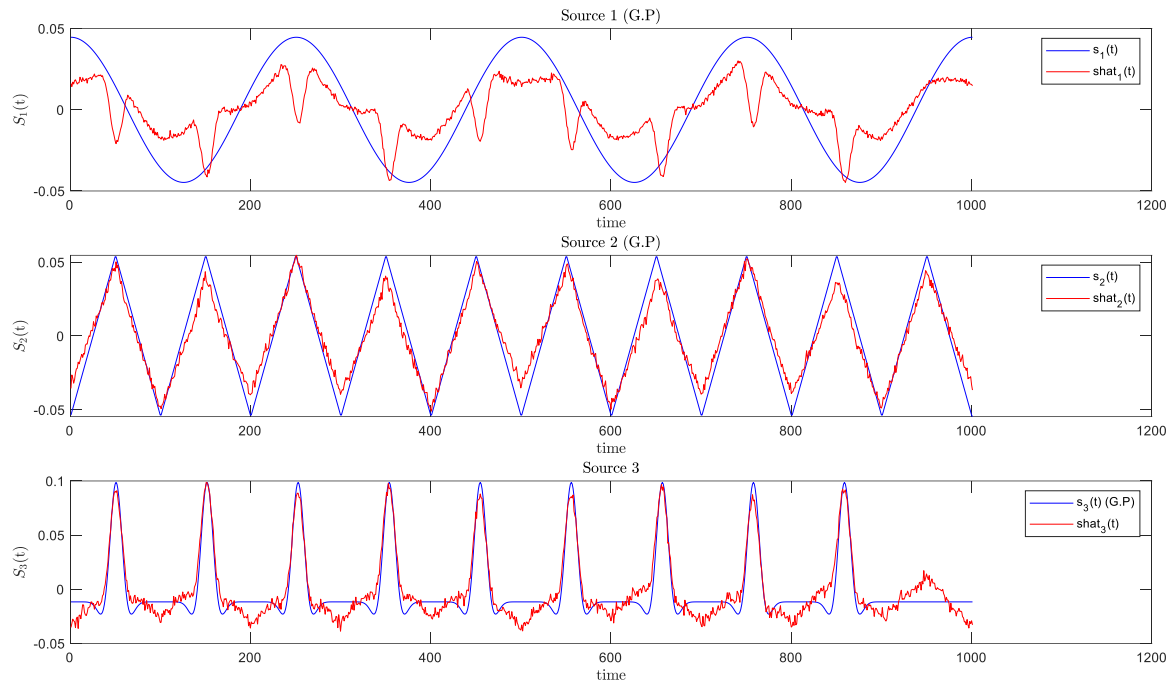
می دهد. بخاطر وجود نویز مقادیر کمی از حالت ایده آل فاصله دارد اما می توان عنصر غیر صفر را تشخیص داد.

Permutation_GP =

0.9640	0.0650	-0.2650
0.1785	0.6090	0.5080
-0.1427	0.9797	-0.9873

تصویر ۲۲: ماتریس $Permutation$

۲. **مقدار خطا** : تصویر ۲۳ ، منابع تخمین زده شده با این روش را نشان می دهد.



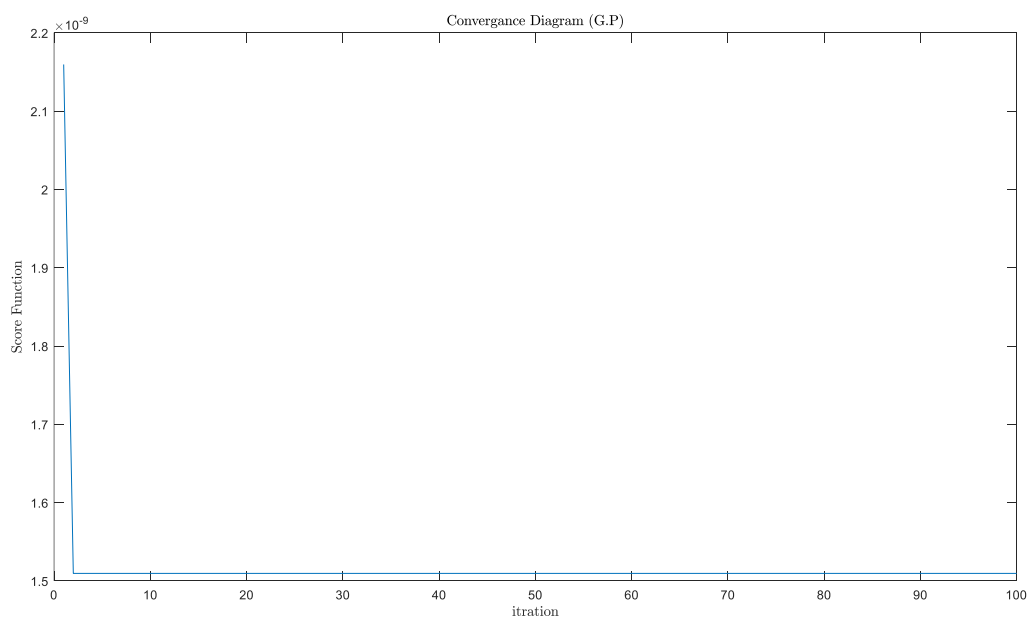
تصویر ۲۳: منابع تخمین زده شده

تصویر ۴ مقدار خطا برای این روش را نشان می‌دهد.

$$G.P : E = 0.362803$$

تصویر ۲۴: مقدار E

۳. نمودار همگرایی تابع هدف: تصویر ۲۵، نمودار همگرایی تابع هدف برای این روش را نشان می‌دهد.



تصویر ۲۵: نمودار همگرایی تابع هدف

سوال ۸.

کد مربوط به روش *FAST ICA* به صورت زیر است.

```
% init
B = [0.321, 0.532, 0.533;
      0.227, 0.41, 0.282;
      0.321, 0.821, 0.81];
B = B ./ sqrt(sum(B.^2, 2));
Y = B * X;
mu = 0.01;
for itr = 1:1000
    Gv = -exp(-v.^2/2);
    y1 = Y(1, :);
    b1 = B(1, :).';
    Gy1 = -exp(-y1.^2/2);
    gy1 = -y1 .* Gy1;
    ggy1 = (y1.^2 - 1) .* Gy1;
    b1 = mean((ones(3, 1) * gy1) .* X, 2) - mean(ggy1) * b1;
    b1 = b1 / norm(b1);

    y2 = Y(2, :);
    b2 = B(2, :).';
    Gy2 = -exp(-y2.^2/2);
    gy2 = -y2 .* Gy2;
    ggy2 = (y2.^2 - 1) .* Gy2;
    b2 = mean((ones(3, 1) * gy2) .* X, 2) - mean(ggy2) * b2;
    b2 = b2 / norm(b2);
    b2 = (eye(3) - b1 * b1.') * b2;
    b2 = b2 / norm(b2);

    y3 = Y(3, :);
    b3 = B(3, :).';
    Gy3 = -exp(-y3.^2/2);
    gy3 = -y3 .* Gy3;
    ggy3 = (y3.^2 - 1) .* Gy3;
    b3 = mean((ones(3, 1) * gy3) .* X, 2) - mean(ggy3) * b3;
    b3 = b3 / norm(b3);
    b3 = (eye(3) - [b1, b2] * [b1, b2].') * b3;
    b3 = b3 / norm(b3);

    B = [b1.'; b2.'; b3.'];
    S_hat = B * X;
    Shat_ = S_hat;
    S_ = S;
    [~, r1] = max(abs(Shat_(1, :) * S_'));
    S_(r1, :) = 0;
    [~, r2] = max(abs(Shat_(2, :) * S_'));
    S_(r2, :) = 0;
    [~, r3] = max(abs(Shat_(3, :) * S_'));
    S_(r3, :) = 0;
    S_hat(r1, :) = Shat_(1, :);
    S_hat(r2, :) = Shat_(2, :);
    S_hat(r3, :) = Shat_(3, :);
    S_hat(1, :) = S_hat(1, :) * max(abs(S(1, :))) / max(abs(S_hat(1, :)));
    S_hat(2, :) = -S_hat(2, :) * max(abs(S(2, :))) / max(abs(S_hat(2, :)));
    S_hat(3, :) = -S_hat(3, :) * max(abs(S(3, :))) / max(abs(S_hat(3, :)));
    KURTY = mean(Y.^4, 2) - 3 * (mean(Y.^2, 2)).^2;
    Score(itr) = norm(KURTY) / L;
    Y = B * Z;
end
figure
plot(Score);
xlim([0 100]);
title('Convergence Diagram (FAST ICA)', 'Interpreter', 'latex');
xlabel('iteration', 'Interpreter', 'latex');
ylabel('Score Function', 'Interpreter', 'latex');
E = norm(S - S_hat, 'fro')^2 / norm(S, 'fro');
fprintf('FAST ICA : E = %f\n', E(end));
```

```
figure
subplot(3,1,1);
plot(t,S(1,:), 'b', t, S_hat(1,:), 'r');
legend('s_1(t)', 'shat_1(t)');
title('Source 1 (FAST ICA)', 'Interpreter', 'latex');
xlabel('time', 'Interpreter', 'latex');
ylabel('$S_1(t)$', 'Interpreter', 'latex');
subplot(3,1,2);
plot(t,S(2,:), 'b', t, S_hat(2,:), 'r');
legend('s_2(t)', 'shat_2(t)');
title('Source 2 (FAST ICA)', 'Interpreter', 'latex');
xlabel('time', 'Interpreter', 'latex');
ylabel('$S_2(t)$', 'Interpreter', 'latex');
subplot(3,1,3);
plot(t,S(3,:), 'b', t, S_hat(3,:), 'r');
legend('s_3(t) (G.P)', 'shat_3(t)');
title('Source 3 (FAST ICA)', 'Interpreter', 'latex');
xlabel('time', 'Interpreter', 'latex');
ylabel('$S_3(t)$', 'Interpreter', 'latex');
Permutation_FAST_ICA = B * A
```

۱. **ماتریس $Permutation$** : این ماتریس را به صورت $B \times A$ بدست می‌آوریم. تصویر ۲۶ این ماتریس را نشان می‌دهد.

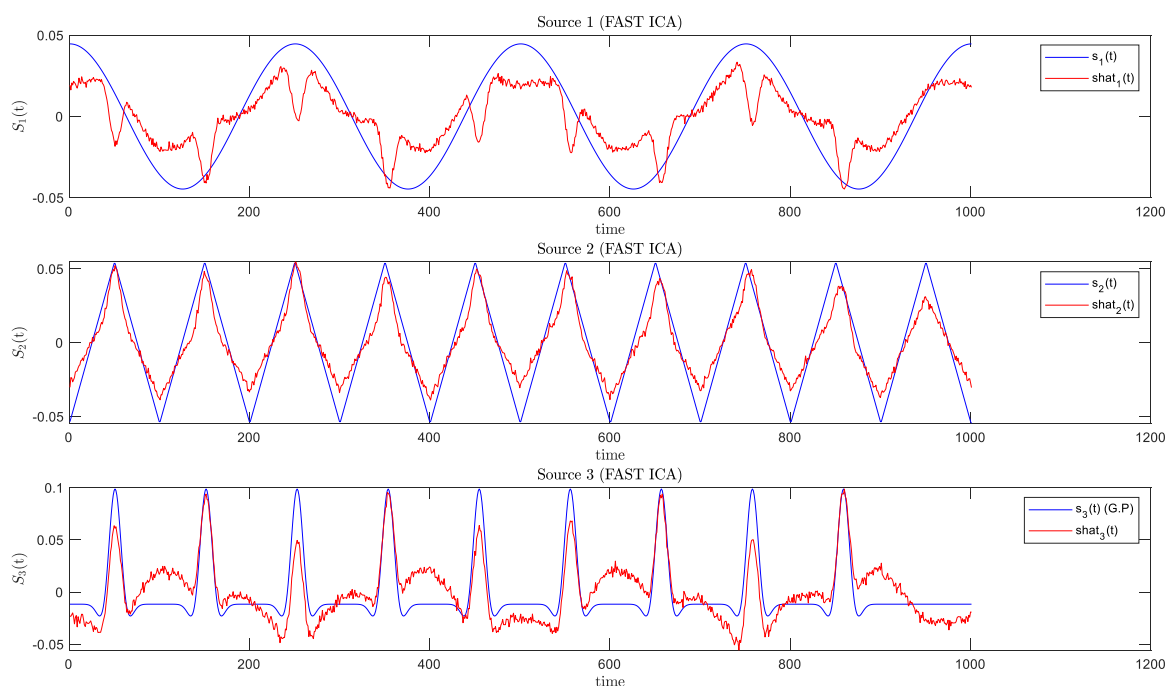
بخاطر وجود نویز مقادیر کمی از حالت ایده آل فاصله دارد اما می‌توان عنصر غیر صفر را تشخیص داد.

Permutation_FAST_ICA =

0.8420	0.2062	-0.5228
-0.1641	-0.9600	-0.3410
0.5138	0.1896	-0.7813

تصویر ۲۶: ماتریس $Permutation$

۲. **مقدار خطا**: تصویر ۲۷، منابع تخمین زده شده با این روش را نشان می‌دهد.



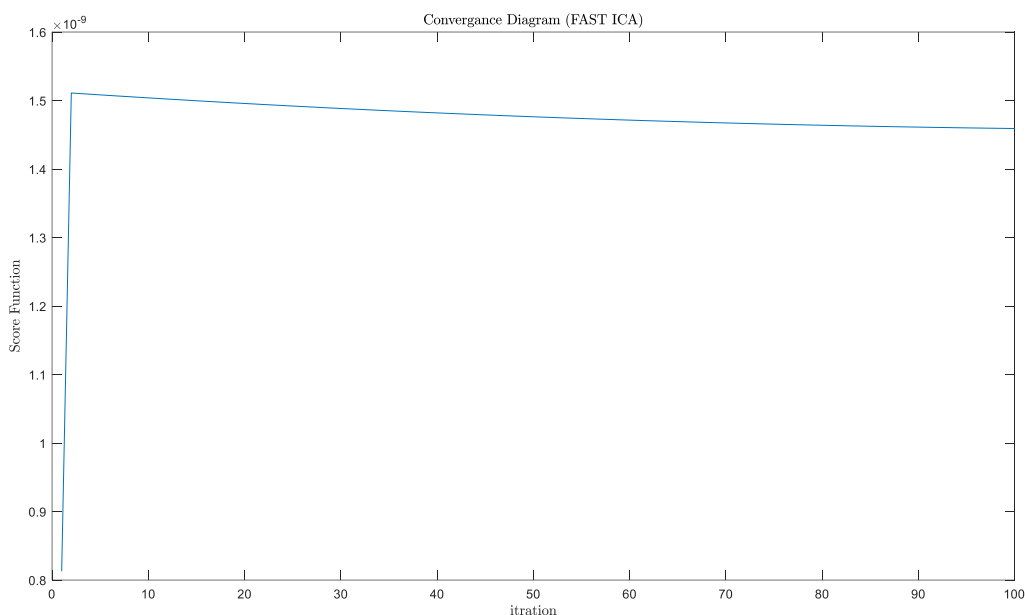
تصویر ۲۷ : منابع تخمین زده شده

تصویر ۲۸ مقدار خطا برای این روش را نشان می‌دهد.

$$\text{FAST ICA} : E = 0.505868$$

تصویر ۲۸ : مقدار E

۳. نمودار همگرایی تابع هدف : تصویر ۲۹ ، نمودار همگرایی تابع هدف برای این روش را نشان می‌دهد.



تصویر ۲۹ : نمودار همگرایی تابع هدف

سوال ۹.

- از نظر سرعت همگرایی :

$$\text{FAST ICA} < G.P = Kurt = \text{fixed} - \text{point}$$

- از نظر کیفیت جداسازی:

$$G.P = Kurt = \text{fixed} - \text{point} < \text{FAST ICA}$$